IA311-Intermediate Report
Skills Similarity Measurement To Enrich An Ontology

Supervisor: M. Mohammad Fallaha
Referent : Mrs. Clavel

Aurélien Chambon
Vathana Thy

January 2021

Contents

1 Introduction

The main purpose of this project is to generate an ontology from a database of skills in order to match job seeker's resume to job descriptions through distances from skills and skills requirements. We have to measure skills similarities using Natural Language Processing models. Then, we must extract skills from job descriptions or resumes in order to calculate distances from their respective skills lists.

2 Intermediate result

2.1 Database overview

Our resource for this project is a database containing skills in two languages: English and French, listed by *name*, *id*, and a list of top 5 to 9 skills associated to this skill, according to the number of profiles where they both appear, named *topSkills*. In this database, we can find around 34000 rows which may contain duplicates, null values and rows referring to the same skill such as "Java" and "Java programming" for instance. Furthermore, skills listed in *topSkills* don't have their *id*, which we have to add in order to link the "skill object" to the string in the *topSkills* list. Here is a look at the first five rows of our original dataset :

	_id	name	topSkills
0	5b057b59b5d2691588302c95	A + Certified	[{'name': 'Certification Network+ ', 'memberCo...
1	5b057b59b5d2691588302c96	A&d	[{'name': 'Pétrole ', 'memberCount': 694}, {'n...
2	5b057b59b5d2691588302c97	A&e	[{'name': 'Services de santé ', 'memberCount':...
3	5b057b59b5d2691588302c98	A&h	NaN
4	5b057b59b5d2691588302c99	A&p	[{'name': 'Aviation ', 'memberCount': 6175}, {...
...	...	...	...

Figure 1: Database overview

2.2 Data cleaning and pre-processing

2.2.1 Handling null values

An important part of every data science project is to have qualitative data. Hence, we spend lots of time cleaning and pre-processing. Our first operation has been to delete every row where *name* has a null value but to keep null *topSkills* as we can still use the relation between *name* and *id* of the skill's row.

2.2.2 Language detection

Before proceeding to another pre-processing step, we need to handle the language of the database. After testing, we found more than 95% of the data are in English. Due to computation time, we are first interested in the case of transforming the whole database into

English. We can optimize this process by translating only the words detected as French into English.

```
There are 13289 French skills (3.903489318203848%) and 327150 English skills (96.09651068179616%).
```

Figure 2: French and English skills proportion

2.2.3 Skills translation

For each skill detected as a French word, we translate them into English and put the skill translated back into the database. We have tried two different tools for translation (GoogleTrans and Google API) and Google API appears to do well its tasks while GoogleTrans is very unstable.

After this step, we have our whole database in English. We then continue to the following step of pre-processing.

2.2.4 Pre-processing : tokenization and stemming

We then had to pre-process our data. Usually, we start by removing accents, superfluous spaces, stopwords and uppercases. We will call this transformation *clean names*. Next, we tokenize sentences and use stemming. Here is a simple example of what we add in the column *clean name* into a *tokenized stemmed clean name* :

```
'modelisation informations construction (bim)'  ⇨  ['modelis', 'inform', 'construct', '(', 'bim', ')']
```

Figure 3: Example of skill tokenization

2.2.5 Searching for duplicates

As you can see from the previous image, this particular skill is a duplicate of the skill **bim**. We had to find such duplicates.

To handle simple duplicates, we simply delete rows where skills' *clean name* are the same. For example: Programmation Java and programmation java. They both have the same clean name.

For a more complicated duplicates, we compare the tokenized skill instead. If the duplicates are found, we then calculate the proportion of common skills in their *topSkills* and delete one of them if this proportion is above 90%.

However, we don't want to lose any information, so we added a new column *synonyms* in which we put the deleted duplicate's *name*. For example, if **modelisation information construction (bim)** has been kept, then **bim** would be in the column *synonym*. This allowed us to delete approximately 3000 duplicates. Below is a look at the dataset at this step of the pre-processing :

	_id	name	topSkills	synonyms
0	5b057b59b5d2691588302c95	{'name': 'A + Certified', 'clean_name': 'a + c...	[{'name': 'Certification Network+ ', 'memberCo...	NaN
1	5b057b59b5d2691588302c96	{'name': 'A&d', 'clean_name': 'a&d', 'tokenize...	[{'name': 'Pétrole ', 'memberCount': 694, 'tok...	NaN
2	5b057b59b5d2691588302c97	{'name': 'A&e', 'clean_name': 'a&e', 'tokenize...	[{'name': 'Services de santé ', 'memberCount':...	NaN
3	5b057b59b5d2691588302c99	{'name': 'A&p', 'clean_name': 'a&p', 'tokenize...	[{'name': 'Aviation ', 'memberCount': 6175, 't...	NaN
4	5b057b59b5d2691588302c9a	{'name': 'A&r Administration', 'clean_name': '...	[{'name': 'Musique ', 'memberCount': 67631, 't...	NaN
...	...	...	...	...
30970	5b0bed10d24d6b2a60d27cd6	{'name': '837i', 'clean_name': '837i', 'tokeni...	[{'name': '837P ', 'memberCount': 494, 'tokeni...	NaN
30971	5b0bed10d24d6b2a60d27cd7	{'name': '837p', 'clean_name': '837p', 'tokeni...	[{'name': '837l ', 'memberCount': 494, 'tokeni...	NaN
30972	5b0bed10d24d6b2a60d27cd9	{'name': '8d Problem Solving', 'clean_name': '...	[{'name': 'Failure Mode and Effects Analysis (...	NaN
30973	5b0bed10d24d6b2a60d27ce2	{'name': '960 Grid System', 'clean_name': '960...	[{'name': 'Cascading Style Sheets (CSS) ', 'me...	NaN
30974	5b0bed10d24d6b2a60d27ce3	{'name': '@task', 'clean_name': '@task', 'toke...	[{'name': 'Project Management ', 'memberCount'...	NaN

30975 rows × 4 columns

Figure 4: Dataset after including synonyms

2.2.6 Adding *id* to our elements in *topSkills*

A laborious work has been to browse the entire dataset to complete each element of topskills with its correspondent skill's *id*. We used *clean names* and only searched in skills with the same first letter (the dataset being ordered alphabetically by *name*). After a few hours of computation, we observed that only 80 percent of the skills in *topSkills* had a correspondent skill in the column *name*. As we were unable to create ids, we had to let elements in *topSkills* without id. Here is an example of a skill in a list of *topSkills* :

```
{'_id': '5b057b59b5d2691588307da1',
 'clean_name': 'microsoft office',
 'memberCount': 7278,
 'name': 'Microsoft Office ',
 'tokenized_skill': ['microsoft', 'office'],
 'tokenized_stem_skill': ['microsoft', 'offic']}
```

Figure 5: Example of a skill in a list of *topSkills*

2.3 Model training

We had the idea of testing some well known models in the community of Natural Language Processing (NLP). Then, we will test the coherence between skills and see which model provides a better result.

2.3.1 Word2Vec

One of the most used model to create word embeddings is Word2Vec, from Google. We decided to give it a try and use every skill concatenated with its *topSkills* to create our sentences. Then, Word2Vec created a vocabulary and give each skill a vector of dimension 200 which we add in the database. In order to check if it worked fine, we tried some simple operations on word embeddings and PCA to represent some distances or clusters.

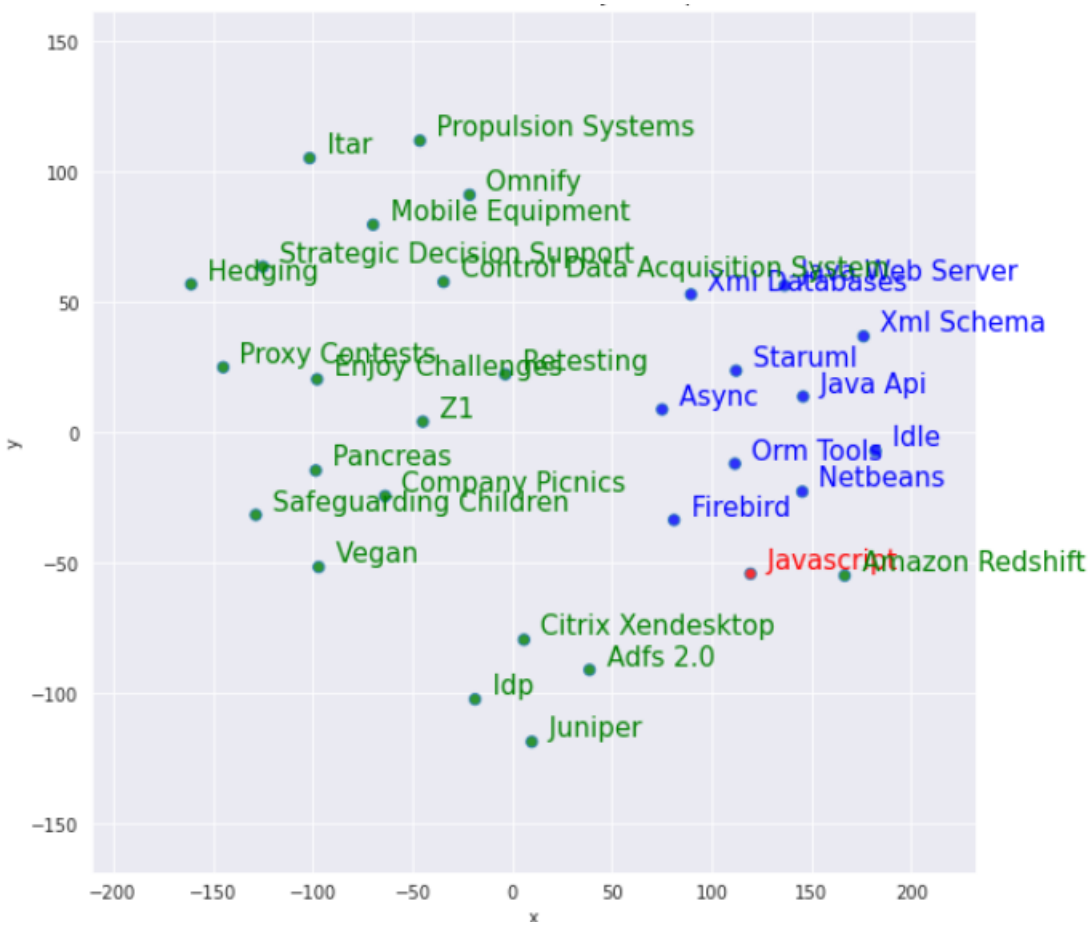

Figure 6: Visualisation of skills that are closed to *JavaScript*. Skills appearing blue are *JavaScript*'s topskills

2.3.2 FastText

FastText is a model introduced by Facebook. After using Word2Vec, we try this model to see whether we can have a better model for word representation.

We plot the figure 7 using Principal Component Analysis (PCA) implemented by *sklearn*. We observe in this example that our model makes sense in terms of similarity between skills. We see that *machine* is around other skills which is in the same category. The same things happen to *artificial intelligence, html and network*. They are skills related to coding. Hence, it is normal to see them in the same cluster. Finally, *massage* is a skill which is completely different compared to other two categories. That is why it is located far away from them, but with other skills similar to it .

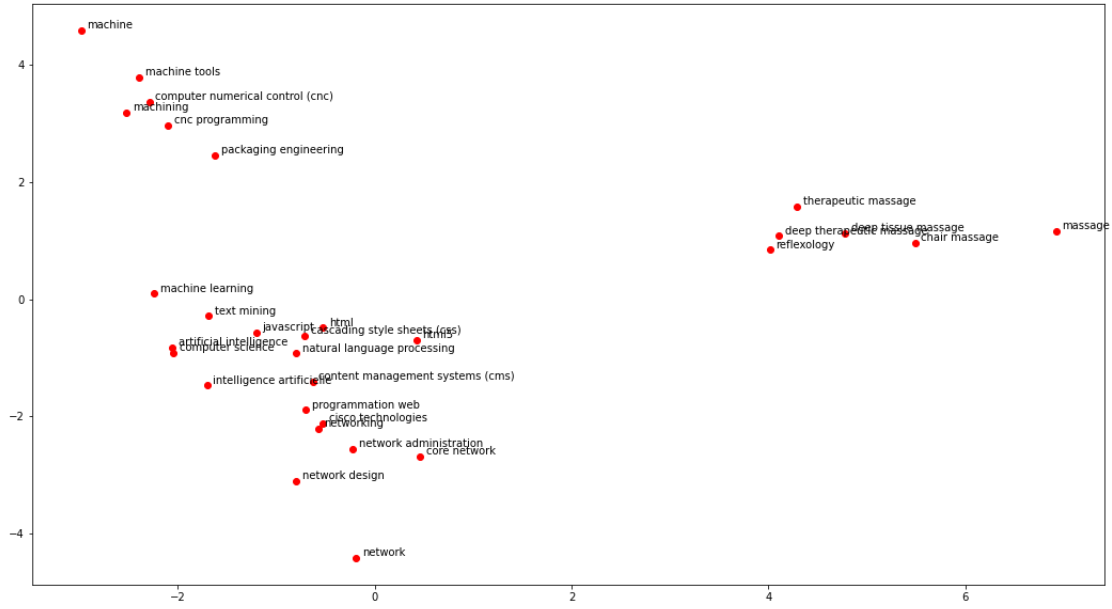

Figure 7: Visualisation of skills that are closed to *artificial intelligence, html, machine, network, and massage.*

3 Difficulties

We have faced many difficulties during this project. The major difficulties encountered are in different step while working on the project.

1. Language detection: we use only one classifier (*Langid*) in order to classify the language of the skills, since other classifiers than this one (NLTK, Googletrans, Pycld3, Langdetect, Chardet, GuessLanguage) don't allow user to specify the desired language to detect (English and French in our case). These classifiers either have very poor and unusable performance or do not work at all. The accuracy of *Langid* is not 100%. However, it is good enough to be used (cf. figure 8).

   The main idea of having many classifiers different is to improve the prediction performance by doing a majority vote using the result given by each classifier. If we base on only one classifier, the result can be undesired. Unfortunately, we are unable to do so, since we have only one classifier (*Langid*) which is reasonable to be used in terms of execution time and result.

2. We have not found a way to relevantly compare the two models trained, as we don't have data to confront them. So, we decided to stay with Word2Vec because the majority of research papers we refer to work on this project use Word2Vec.

	skill	langid	nltk	googletrans	pycld3	langdetect
0	Aviation	en	lms	en	mg	en
1	Maintenance avions	fr	fra	en	ny	fr
2	Compagnie aérienne	fr	fra	en	fr	fr
3	Aérospatiale	fr	por	en	fr	fr
4	Navigabilité	fr	lld	en	fr	fr
5	Aviation commerciale	fr	lms	en	en	it
6	Sécurité en vol	fr	fra	en	fr	fr
7	Avionique	fr	lnc	en	la	fr
8	Aéronautique	fr	fra	en	fr	fr
9	Systèmes d'avion	fr	fra	en	fr	fr

Figure 8: Comparison of different classifier's prediction

4 Project planning

Our project planning is actually to focus on the three main features we have to implement :

1. Understand and apply the process of finding duplicates using Wikipedia pages.
2. Extract skills from a job description: a resume, a text ...etc.
3. Finding a way to compute the distance between a resume and a job description. Also, we could look further into models comparison.